(12) UK Patent Application (19) GB (11) 2 238 215 (13) A

(43) Date of A publication 22.05.1991

(21) Application No 9023830.4

(22) Date of filing 02.11.1990

(30) Priority data
(31) 435553 (32) 13.11.1989 (33) US

(71) Applicant
Apple Computer Inc

(Incorporated in the USA - California)

20525 Mariani Avenue, Cupertino, California 95014, United States of America

(72) Inventor
Daniel S Venolia

(74) Agent and/or Address for Service
Potts Kerr and Co
15 Hamilton Square, Birkenhead, Merseyside, L41 6BR, United Kingdom

(51) INT CL[5]
G09G 1/00, G06F 15/72

(52) UK CL (Edition K)
H4T TBEX TBLC T126 T130 T149

(56) Documents cited
None

(58) Field of search
UK CL (Edition K) H4T TBBA TBBD TBBX TBEA TBEX TBLC TBLM TBLX
INT CL[5] G06F, G09G
On-line databases: WPI, INSPEC, CLAIMS

(54) Computer display system with a three dimensional cursor shadow

(57) The system provides for the display of a shadow region 3 associated with a three dimensional cursor 2. A cursor control device is used to move the cursor in three dimensions on the display screen, wherein the shadow region is displayed on a shadow plane 4 and follows the movement of the cursor. A shadow represented by an opaque face cursor or a translucent face cursor is provided. The system can detect situations where either the cursor 2 or the shadow 3 are not visible.

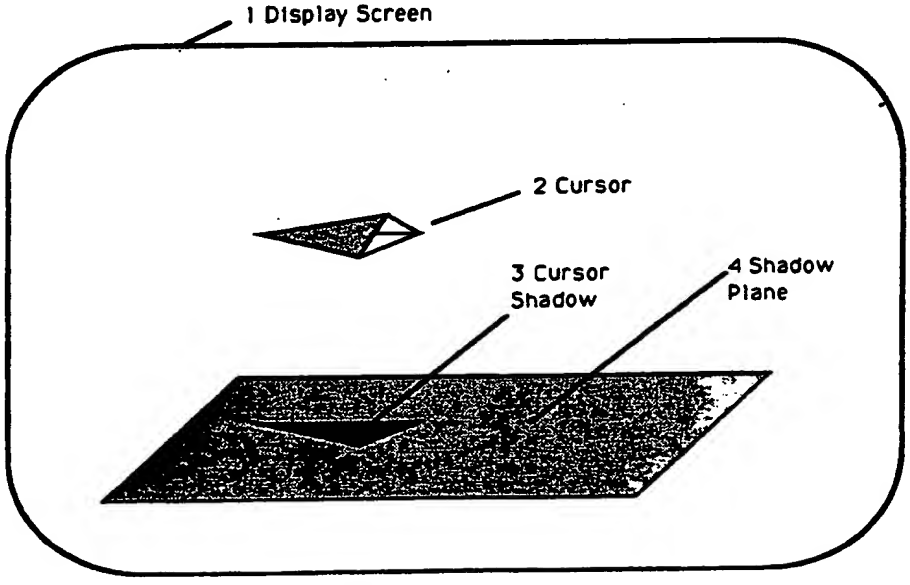


FIG. 1

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

GB 2 238 215 A

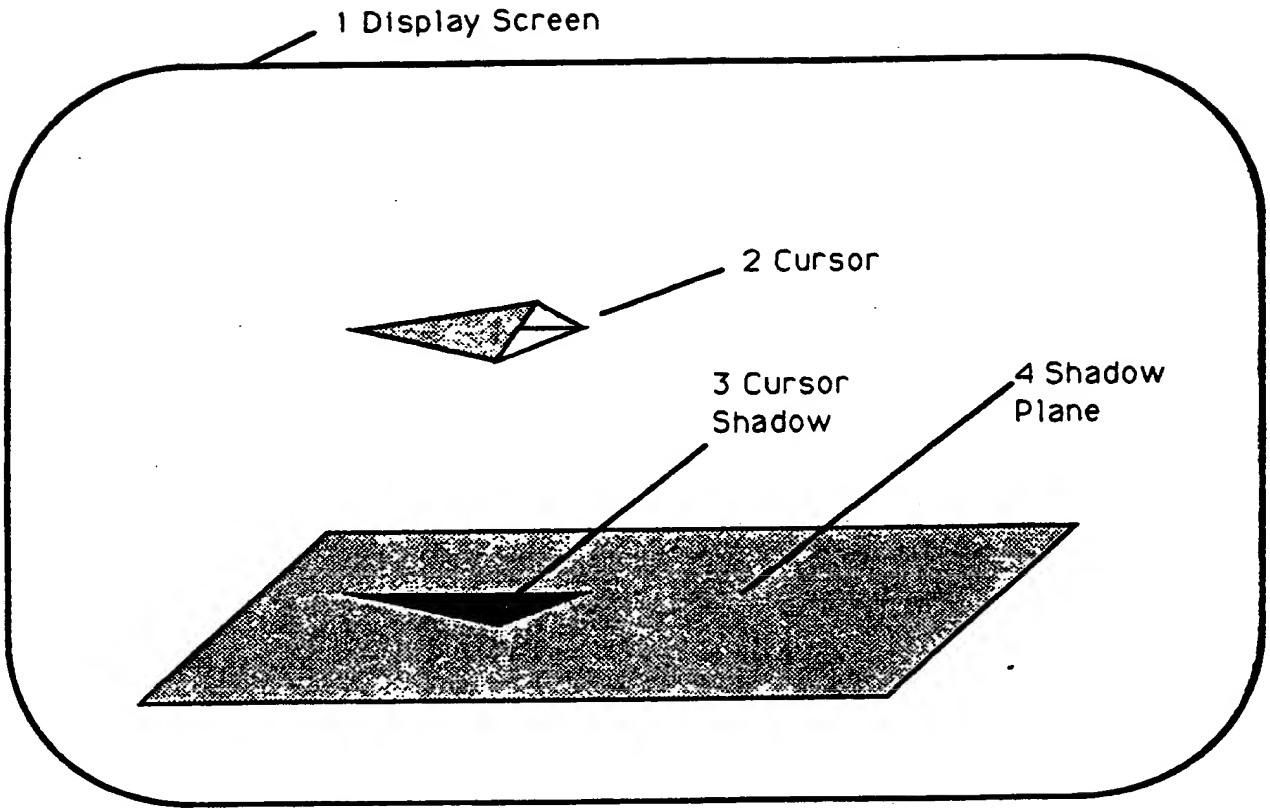


FIG. 1

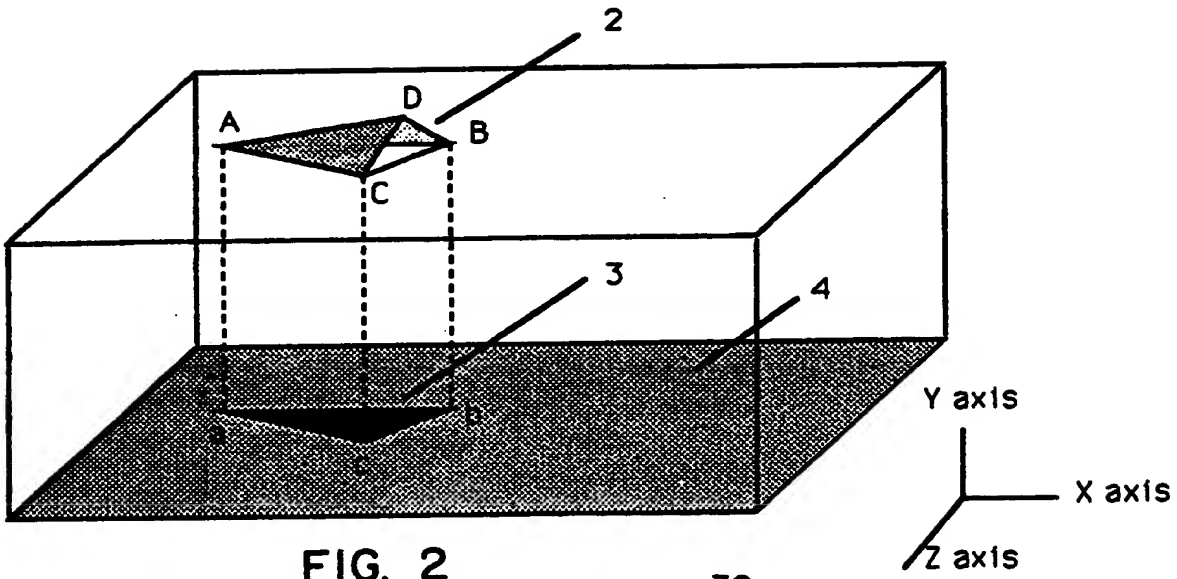


FIG. 2

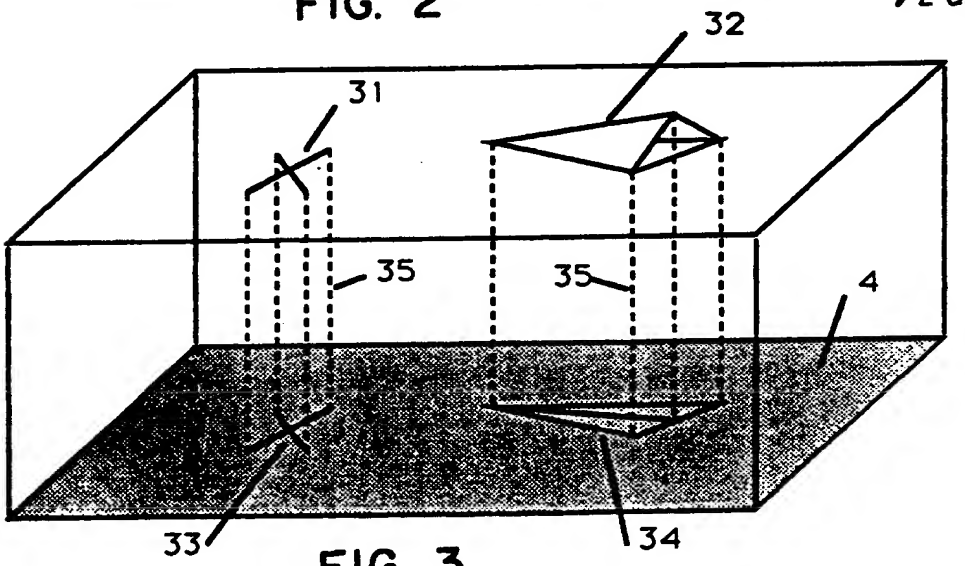


FIG. 3

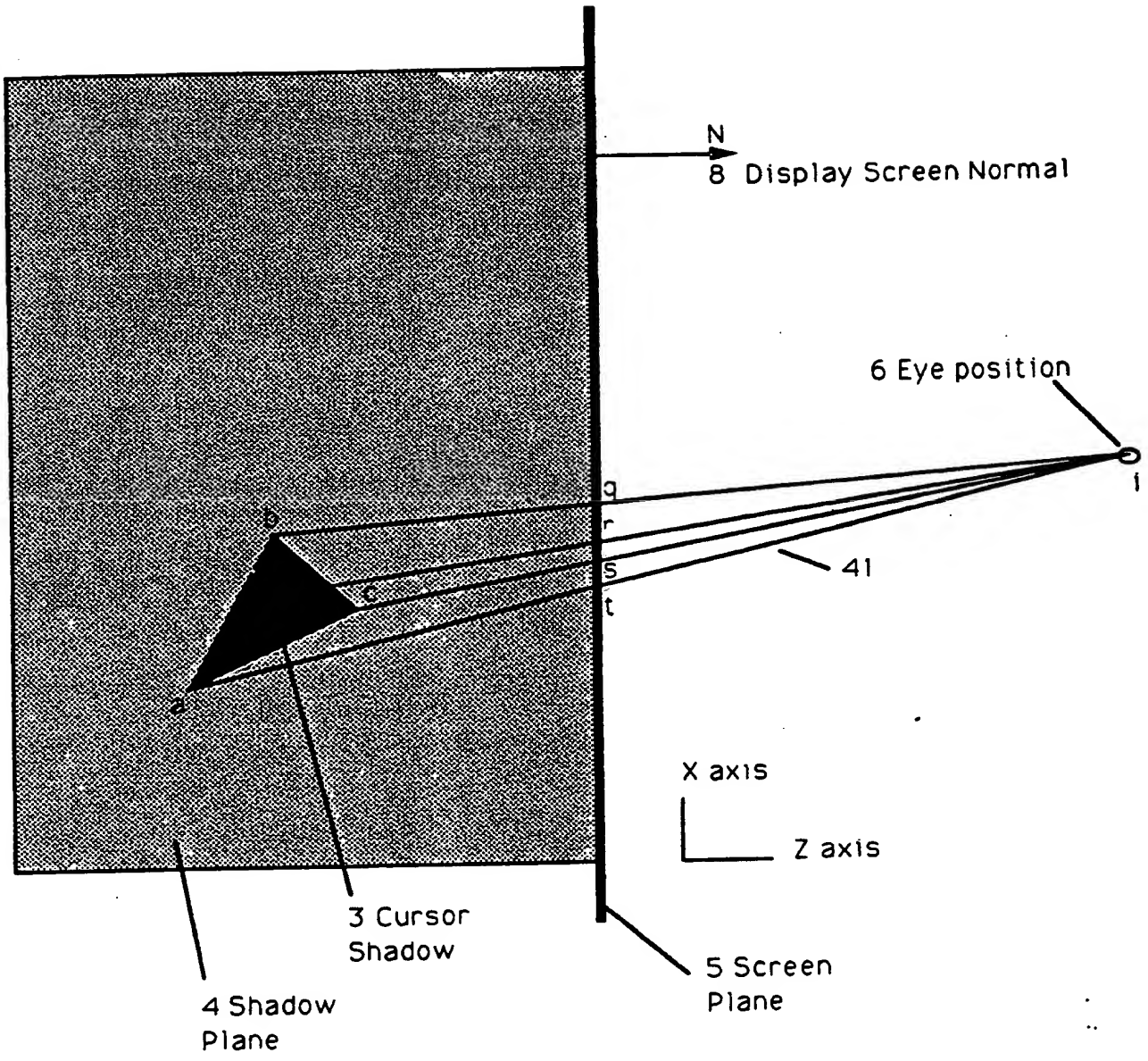

N
8 Display Screen Normal
6 Eye position
q
r
s
t
1
41
b
c
a
X axis
Z axis
3 Cursor
Shadow
4 Shadow
Plane
5 Screen
Plane

FIG. 4

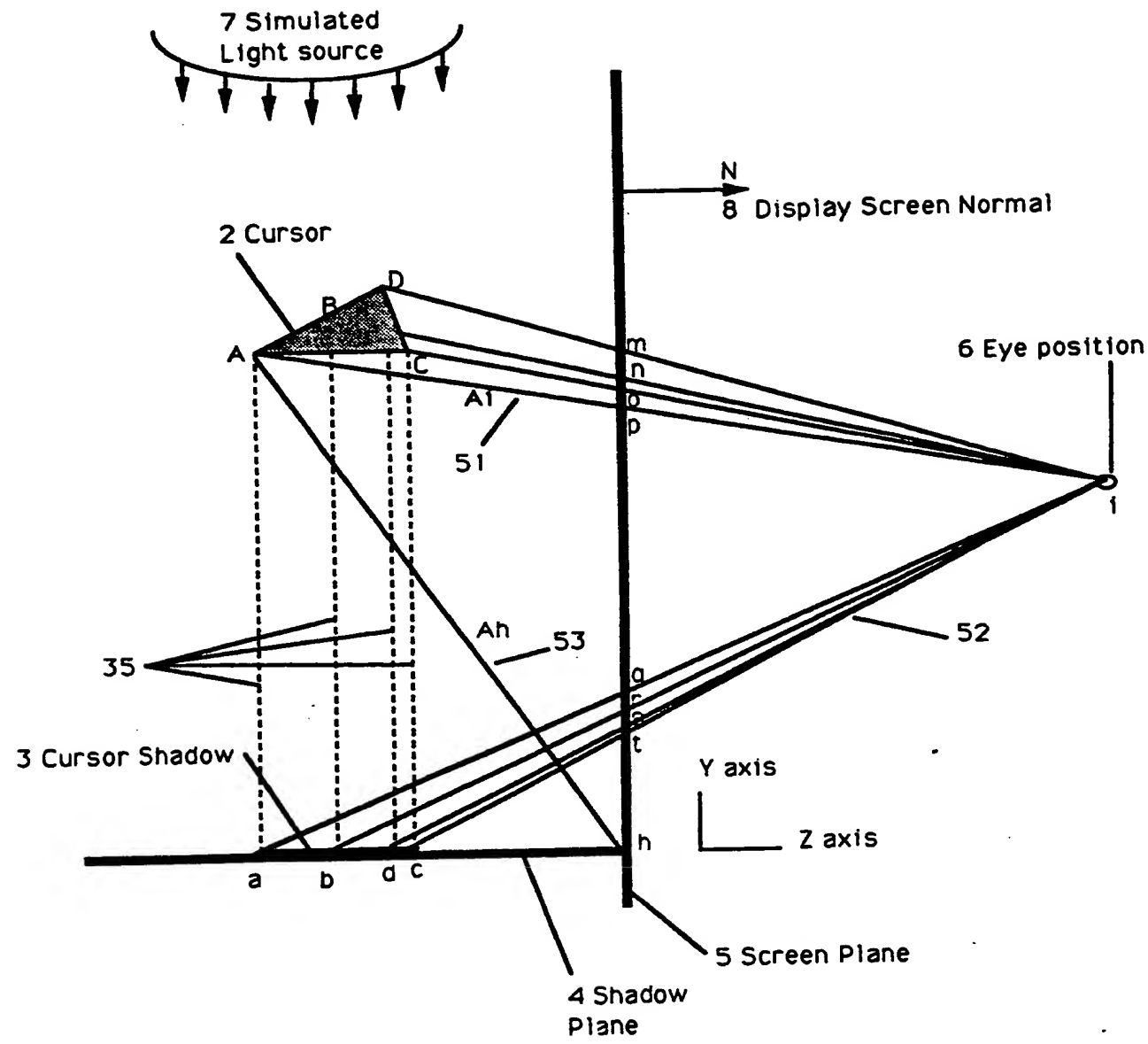

7 Simulated
Light source
N
8 Display Screen Normal
2 Cursor
D
B
A
C
m
n
o
p
6 Eye position
Δl
51
1
Δh
53
52
35
q
r
s
t
3 Cursor Shadow
Y axis
Z axis
h
a
b
d
c
5 Screen Plane
4 Shadow
Plane

FIG. 5

FIG. 6

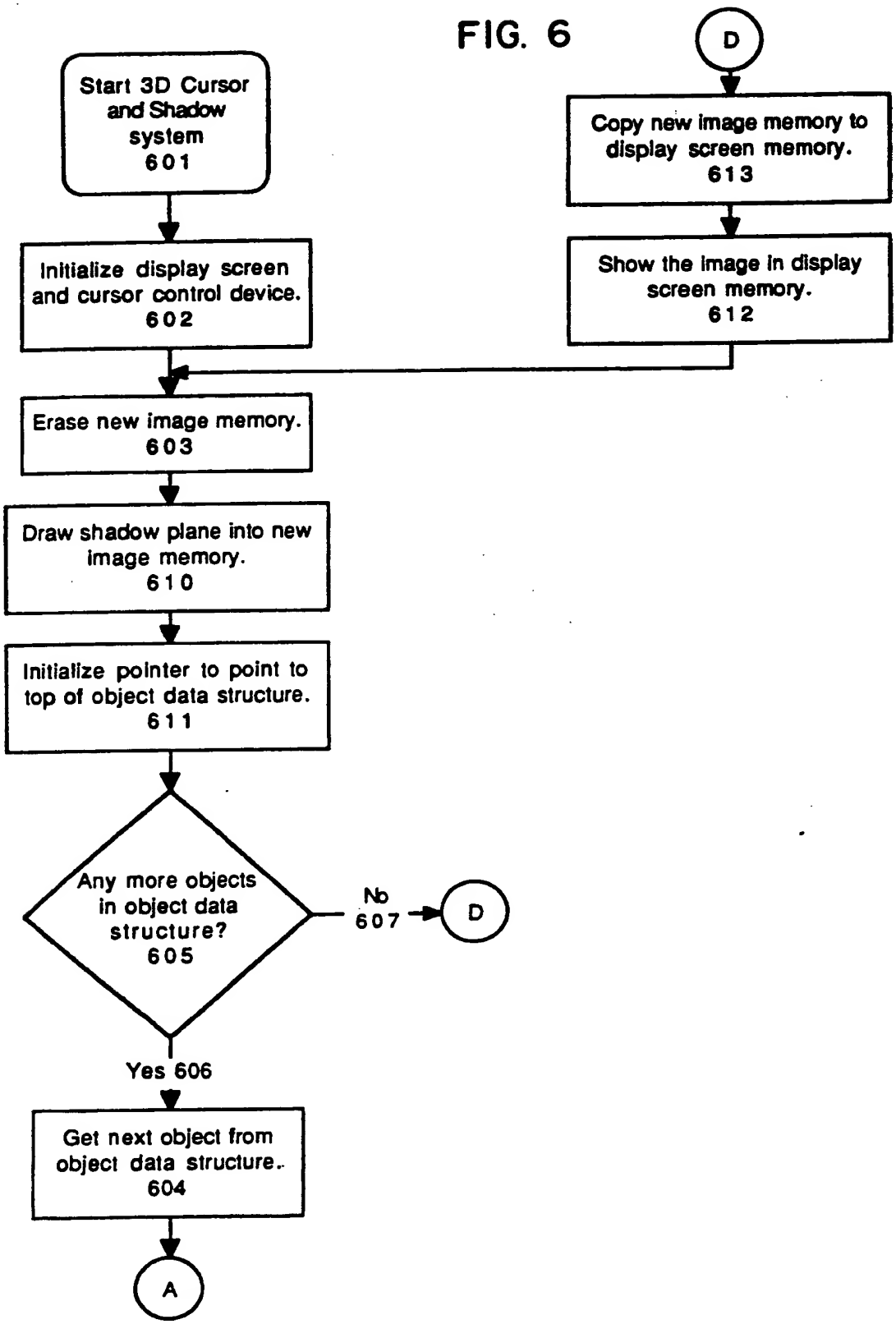

Start 3D Cursor and Shadow system 601
Initialize display screen and cursor control device. 602
Erase new image memory. 603
Draw shadow plane into new image memory. 610
Initialize pointer to point to top of object data structure. 611
Any more objects in object data structure? 605
No 607
D
Yes 606
Get next object from object data structure. 604
A
D
Copy new image memory to display screen memory. 613
Show the image in display screen memory. 612

FIG. 7

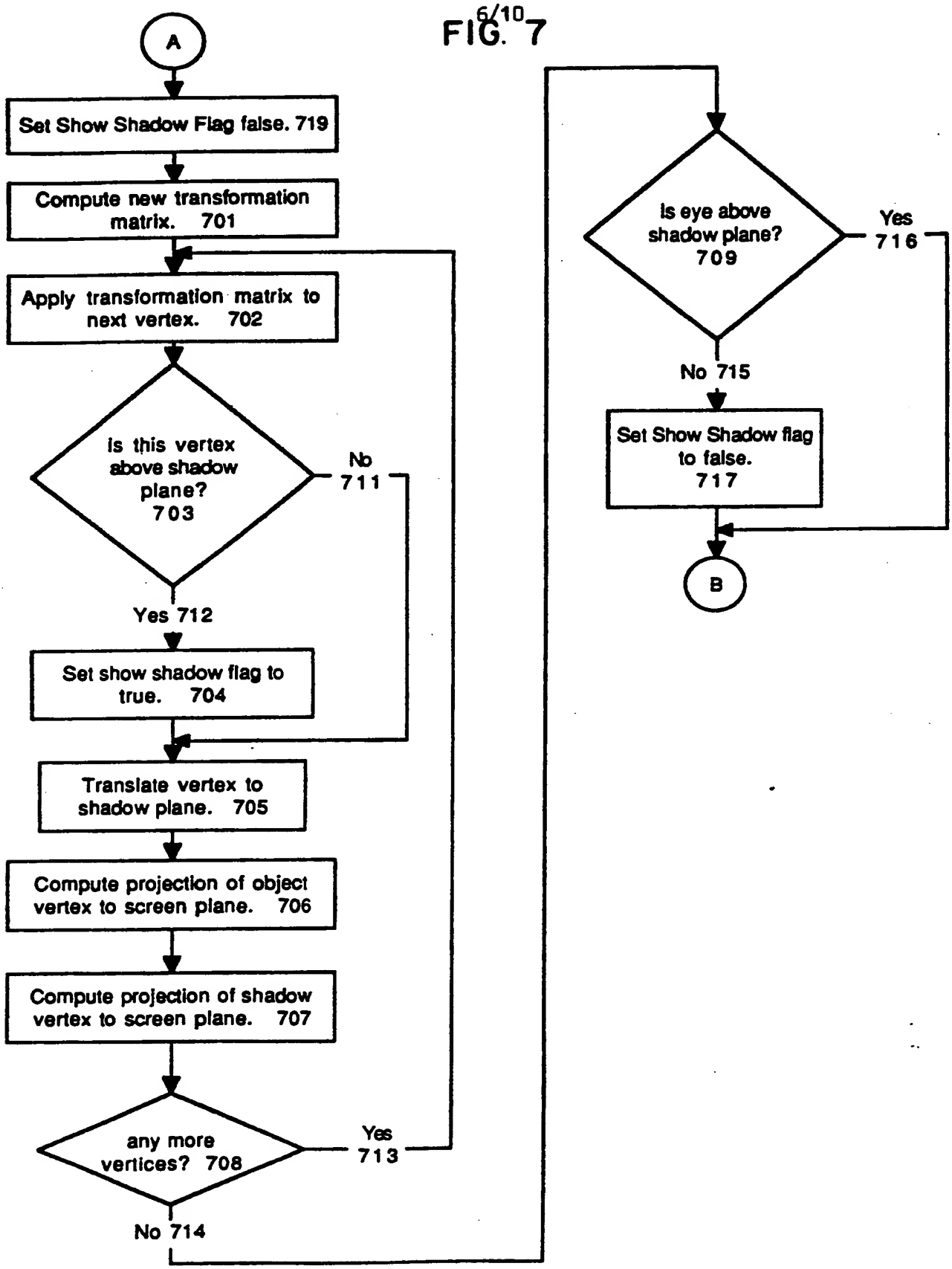

A
Set Show Shadow Flag false. 719
Compute new transformation matrix. 701
Apply transformation matrix to next vertex. 702
Is this vertex above shadow plane? 703
No 711
Yes 712
Set show shadow flag to true. 704
Translate vertex to shadow plane. 705
Compute projection of object vertex to screen plane. 706
Compute projection of shadow vertex to screen plane. 707
any more vertices? 708
Yes 713
No 714
Is eye above shadow plane? 709
Yes 716
No 715
Set Show Shadow flag to false. 717
B

FIG. 8

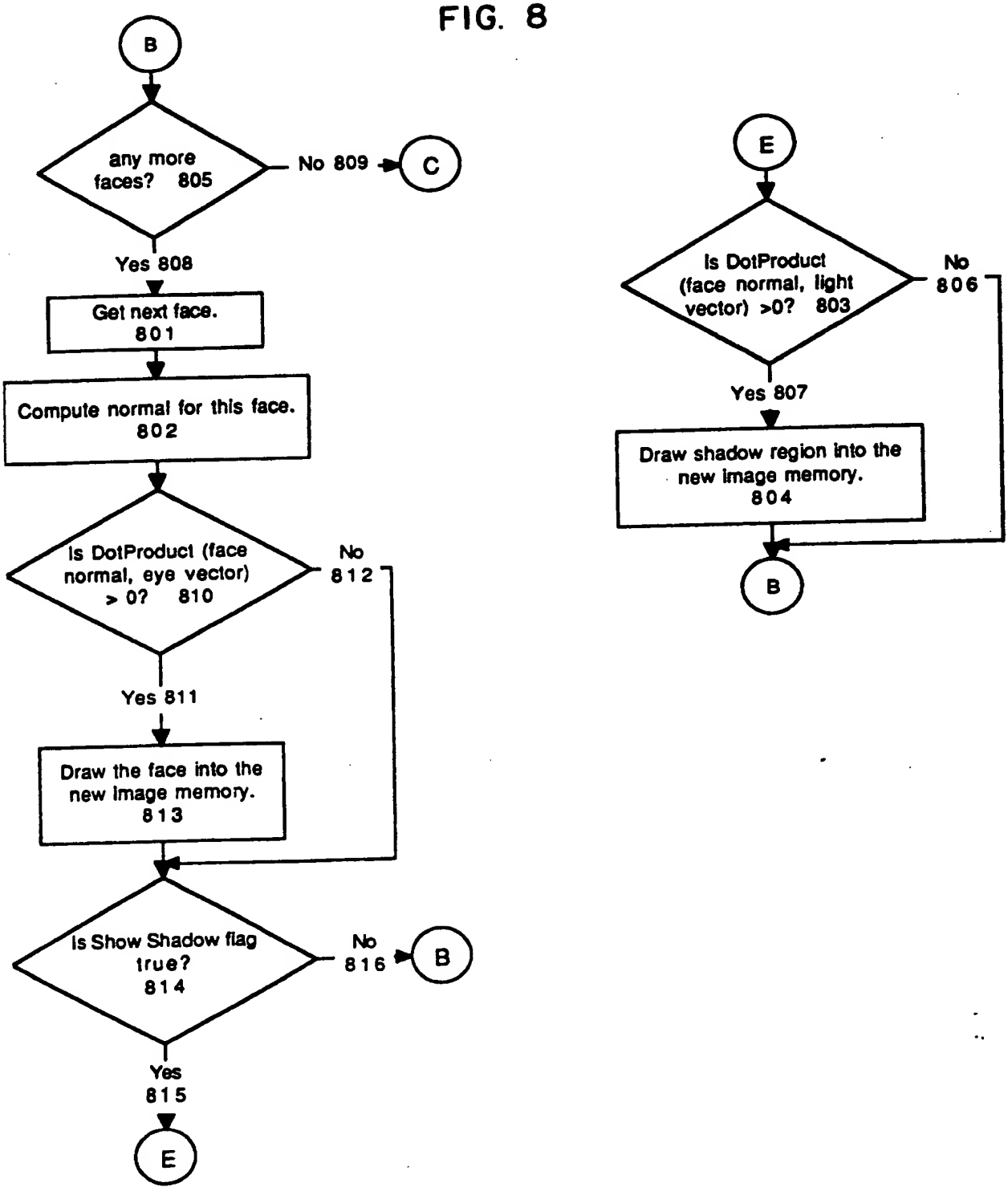
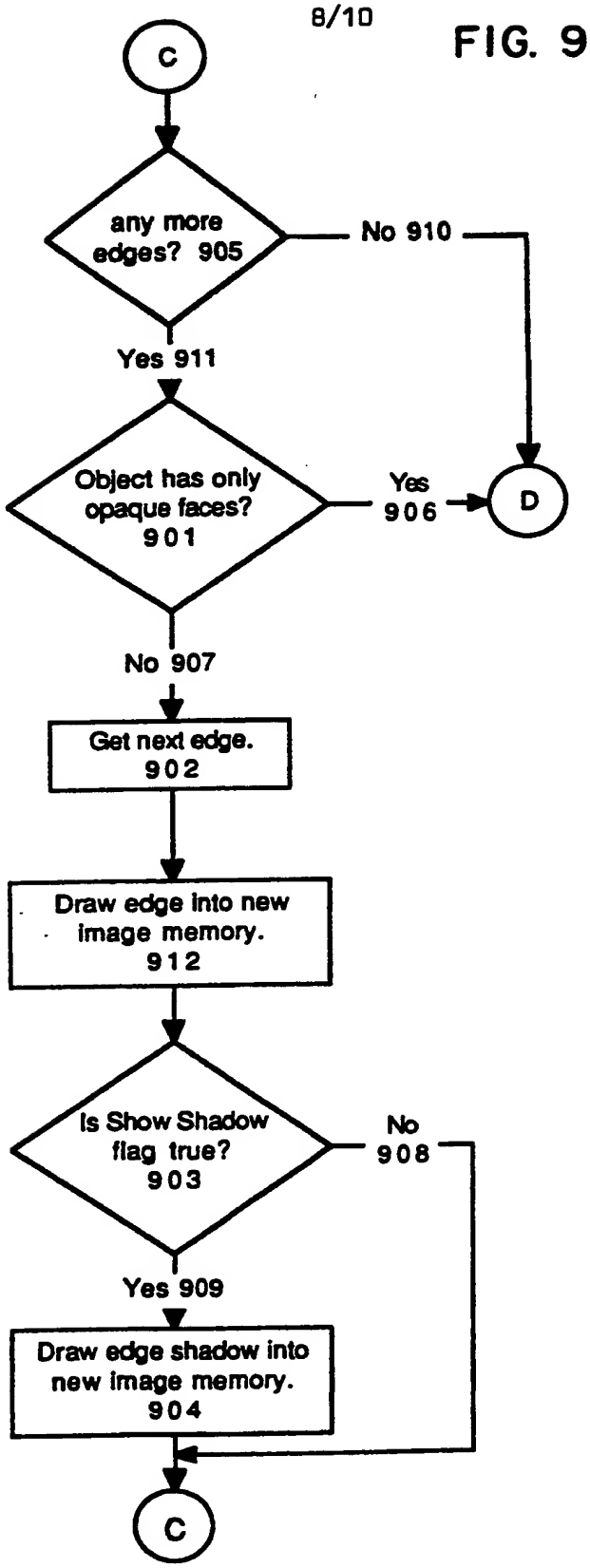

B
any more faces? 805
No 809
C
Yes 808
Get next face. 801
Compute normal for this face. 802
Is DotProduct (face normal, eye vector) > 0? 810
No 812
Yes 811
Draw the face into the new image memory. 813
Is Show Shadow flag true? 814
No 816
B
Yes 815
E
E
Is DotProduct (face normal, light vector) >0? 803
No 806
Yes 807
Draw shadow region into the new image memory. 804
B

FIG. 9


C
any more edges? 905
No 910
Yes 911
Object has only opaque faces? 901
Yes 906
D
No 907
Get next edge. 902
Draw edge into new image memory. 912
Is Show Shadow flag true? 903
No 908
Yes 909
Draw edge shadow into new image memory. 904
C

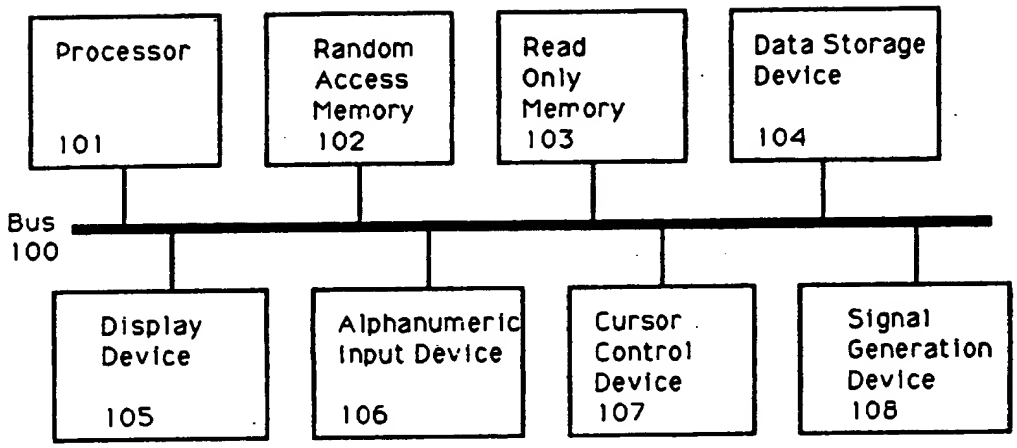

Processor
101
Random Access Memory
102
Read Only Memory
103
Data Storage Device
104
Bus
100
Display Device
105
Alphanumeric Input Device
106
Cursor Control Device
107
Signal Generation Device
108

FIG. 10

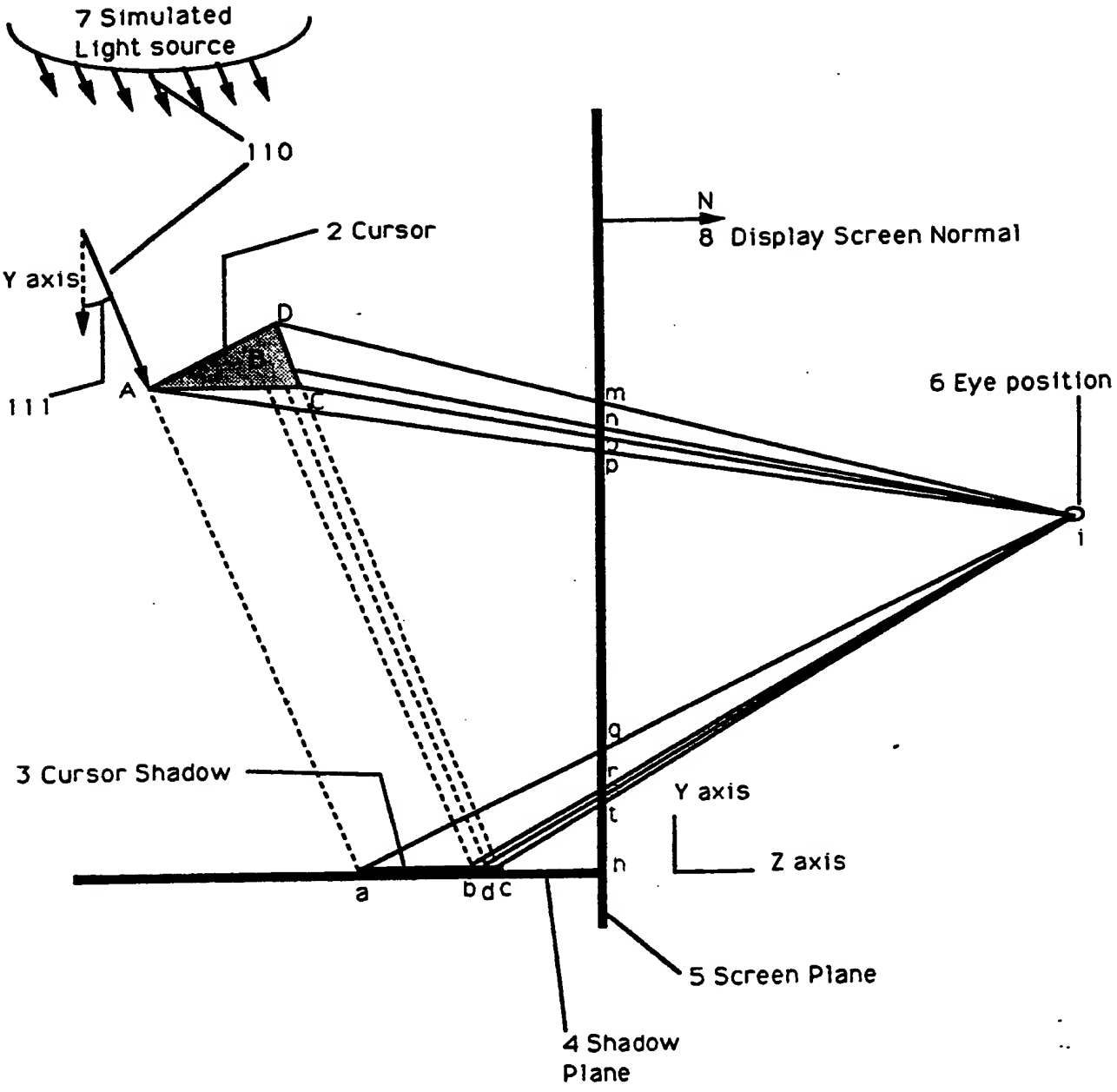

7 Simulated
Light source
110
2 Cursor
N
8 Display Screen Normal
Y axis
111
A
B
D
C
6 Eye position
m
n
o
p
i
q
r
t
n
3 Cursor Shadow
Y axis
Z axis
a
b d c
5 Screen Plane
4 Shadow
Plane

FIG. 11

METHOD AND APPARATUS FOR A COMPUTER DISPLAY SYSTEM WITH A THREE DIMENSIONAL CURSOR SHADOW

(1) Field of the Invention

This invention relates to computer systems having a three dimensional cursor which is displayed on a computer controlled display device. Specifically, the field of this invention is that of interactive computer systems having a cursor capable of representing three dimensional movement where the cursor may be moved on the display screen.

(2) Prior Art

Computer-controlled display devices are useful for the representational display of three dimensional (3D) objects in three dimensional spatial relationships. This is especially true when the displayed objects can be moved around within the 3D space using a cursor control device. Unfortunately, the modeling of 3D space to a display screen (e.g. a CRT computer monitor) is difficult, since the screen is only a two dimensional plane. In prior art, the two dimensional screen limitation creates confusing display images for the user attempting to move a cursor in 3D space. The depth and height of objects is not easy to ascertain without some reference point. The relationship of one object to another is also difficult to interpret, especially as objects move around in space. Prior art three dimensional modeling techniques do not employ shadowing of objects in conjunction with a cursor, having an associated shadow region, as a means for assisting the user in distinguishing the movement and position of objects in space. These problems have hindered the proliferation of computer display devices as a means for displaying three dimensional spaces and the manipulation of a cursor therein.

SUMMARY OF INVENTION

The present invention provides a device and method of facilitating the three dimensional movement of objects on a display screen by displaying a shadow region associated with a cursor. The shadow region assists the user in orienting the relative position and movement of objects within the 3D space. The shadow region may be denoted as a drop shadow, since the shadow region in the preferred embodiment is displayed on a shadow plane located toward the bottom of the display screen.

The present invention comprises an interactive computer-controlled display system including a bus for interconnecting system components, a processor, a random access memory, a read only memory, a data storage means for storing data, a data display device including a display screen, an alphanumeric input device, a cursor control device for interactively positioning a cursor on a display screen, and a signal generation device. This invention provides a means and method for displaying a shadow region in association with a three dimensional cursor whereby the shadow changes shape (including size) and orientation in response to the 3D movement of the cursor device.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is an illustration of the three dimensional cursor with the shadow as displayed on a display screen after activation.

Figure 2 is an illustration of the three dimensional cursor and the shadow with the vertices and axes defined.

Figure 3 is an illustration of the three dimensional cursor and the shadow portrayed as a wire frame object.

Figure 4 is an illustration of the shadow in the X-Z plane.

Figure 5 is an illustration of the shadow and cursor in the Y-Z plane.

Figures 6,7,8 and 9 are flowcharts of the representative computer program that implements this scheme.

Figure 10 is an illustration of the computer system architecture.

Figure 11 is an illustration of the shadow and cursor in the Y-Z plane with the light source displaced at an angle to the Y axis.

DETAILED DESCRIPTION OF THE INVENTION

The present invention provides a device and a process for generating and displaying a shadow region associated with a three dimensional cursor which is displayed on a display device. The shadow region represents the shadow of the 3D cursor and moves with the cursor as the cursor is moved on the display device.

The preferred embodiment of the present invention is implemented on an Apple Macintosh™ computer system. It is obvious to one of ordinary skill in the art, however, that alternative systems may be employed. In general, such systems as illustrated by Figure 10 comprise a bus 100 for communicating information, a processor 101 coupled with said bus for processing information, a random access memory 102 coupled with said bus 100 for storing information and instructions for said processor 101, a read only memory 103 coupled with said bus 100 for storing static information and instructions for said processor 101, a data storage device 104 such as a magnetic disk and disk drive coupled with said bus 100 for storing information and instructions, a display device 105 coupled to said bus 100 for displaying information to the computer user, an alphanumeric input device 106 including alphanumeric and function keys coupled to said bus 100 for communicating information and command selections to said processor 101, a cursor control device 107 coupled to said bus for

communicating information and command selections to said processor 101, a signal generation device 108 coupled to said bus 100 for communicating command selections to said processor 101.

The display device may be a liquid crystal device, cathode ray tube, or other suitable display device. The display device should be able to display at least discrete points and vectors visible on the display screen. More sophisticated display devices can display regions on the display screen in different colors or shades of gray.

The cursor control device 107 is an interactive device with which the computer user may dynamically signal the movement of a visible object on a display screen in three dimensions. Many implementations of a 3D cursor control device are known in the art including three axis mechanical devices (e.g. a trackball) and three dimensional magnetic field detection and positioning systems. Other types of cursor control devices can signal movement in only two dimensions. These two-dimensional cursor control devices, when used in conjunction with another signal means, can be used to provide three dimensional signals. The other signal means may be generated by the alphanumeric keyboard 106 or the signal generation device 108. The alphanumeric keyboard 106 by itself could also be used to produce three dimensional signals. Specific keys on the keyboard could be assigned to signal movement in each of the three dimensions. These cursor control devices typically provide data to the processor in terms of cursor movement in the X, Y, and Z directions.

In this preferred embodiment, the X, Y, and Z directions will be defined as follows: the origin of the screen is defined as the lower left corner of the display screen; X represents an axis extending orthogonally from the left side of the display screen to the right side of the screen parallel to the surface of the screen; the Y axis

extends orthogonally from the bottom of the display screen to the top of the screen parallel to the surface of the screen perpendicular to the X axis; the Z axis extends perpendicular to the display screen from the surface of the screen pointing outward directly toward the user viewing the screen perpendicular to the X and Y axes.

The features and processes of the three dimensional cursor shadow in a preferred embodiment are described below. The general operation of cursor control devices or display systems is described only where necessary to provide a thorough understanding of the present invention, since these methods are already known to those skilled in the relevant art.

Operation of the Present Invention

The cursor shadow 3, shown in Figure 1, is a displayed image associated with a 3D cursor which moves with the cursor control device 107. The two dimensional shape of the shadow corresponds to the three dimensional shape of the cursor 2. In the present invention, the cursor 2 is displayed as a 3 D object. A 3D object is a solid defined using vertices positioned in three dimensional space connected by straight edges. The edges combine to create polygons that define the faces of the solid. For each polygon face, the edges that comprise the face are coplanar; thus, each face has an associated normal vector which is perpendicular to the face and each of the edges therein. Each edge and each face may have an associated color, shade of gray, or display intensity. The surface of each face may be opaque or translucent. As depicted in Figure 2, an object created with opaque faces 2 would cast a homogeneous shadow 3 for each of the opaque faces exposed to a light source 7 (shown in Figure 5). An object created with translucent faces 32

would cast a shadow 34 only at its edges; this type of object appears as a wire frame solid 32 as shown in Figure 3.

The 3D cursor object may also be displayed as a 3D non-solid object 31. That is, the object does not have faces or an interior region. This type of object is displayed as a group of vectors. Just like the solid object, the vector group moves in response to the movement of the cursor control device 107. A cross hair 31, shown in Figure 3, is an example of the non-solid type of cursor object. The non-solid cursor object is represented in the present invention as a group of 3D points connected by vectors. There are no associated polygons or faces. In the case of a cross hair cursor 31, the object can be represented by four points, two for each of the vectors comprising the cross hair. If the points are denoted as vertices and the vectors are denoted as edges, the non-solid object can be represented in a similar manner as the solid object, except that faces do not need to be processed. As in the solid object, each edge (vector) may have an associated color, shade of gray, or intensity. A non-solid object casts a shadow 33 only at its edges.

Any 3D object in the domain of the present invention can be represented in a data structure as a collection of vertices, faces, and edges. The data structure can be stored in computer memory 102, on a magnetic storage media 104, or other suitable data storage means. For each of the object vertices, the X, Y, and Z position of the vertex is saved in the data structure. For each of the object faces, the data retained in the object data structure includes, but is not limited to the following: the number of vertices in the face, the pointer to each associated vertex, the color of the face, and the magnitude and direction of the vector normal to the face. For each edge, the data saved includes, but is not limited to: the pointer to the start and end vertices of the edge, a pointer to the faces to

which this edge is a member, and the color or intensity of the edge. For a non-solid object, the same object data structure can be used; although, the object has only vertices and edges, but no faces. Since the cursor is an object, object data associated with the cursor (cursor data) is also stored in the object data structure.

Each object and the shadow of each object is displayed on the display screen 1 with reference to an eye position 6. The eye position 6, shown in Figures 4 and 5, is a known point in space in front of the display screen. The eye position 6 is a point from where the user would typically view the displayed images. In the present embodiment, this position is defined as a point in 3 D space denoted l(x,y,z) as shown in Figures 4 and 5. The eye position 6 is not necessarily a fixed point. An equivalent embodiment could be conceived whereby the eye position 6 is located in a position different from the position depicted in Figures 4 and 5. The implementation described herein can be used for embodiments employing a variable eye position.

Each image is displayed to the user as a projection from three dimensional space to a two dimensional screen plane 5. The screen plane 5, which is typically the screen of a CRT display device, is perpendicular to the Z axis as shown in Figures 4 and 5. The projection is used to determine the point in the screen plane 5 intersecting a line connecting each object vertex with the eye position 6. For example, as shown in Figure 5, a point p is the intersection of line Ai 51 with the screen plane 5. Line Ah 53 connects the object vertex with the coordinate space origin (h). A vector N 8 is a normal vector perpendicular to the screen plane 5. The normal N is computed using the cross product with two distinct unit vectors in the screen plane, a technique well known in the art. The location of the screen plane intersection point p is generated

first by calculating the dot product of the vector Ah 53 with the screen normal vector N. Next, the dot product is computed for the vector Ai 51 with the screen normal vector N. The quantity of the dot product for Ah divided by the dot product for Ai is multiplied by the X,Y components of the Ai vector and added to the coordinate components of the object vertex. The result is the point p in the screen plane 5 intersecting the line from the object vertex to the eye position 6. The projection for each object vertex is computed as described in the following formulas for cursor vertex A:

$$p(x) = (\text{DotProduct}(N, Ah) / \text{DotProduct}(N, Ai)) * Ai(x) + A(x)$$

$$p(y) = (\text{DotProduct}(N, Ah) / \text{DotProduct}(N, Ai)) * Ai(y) + A(y)$$

$$p(z) = \text{Screen Plane}(z)$$

The position of each of the other object vertices as projected to the screen plane 5 is computed in similar fashion and stored in the object data structure. Other means exist for projecting a vertex to a plane. The means described above is only one satisfactory method used in the present embodiment.

The cursor 2 changes position and orientation on the display screen 1 in response to the three dimensional movement of the cursor control device 107. The cursor control device 107, coupled to the processor 101 via bus 100 as shown in Figure 10, provides signals to the processor 101 indicating the movement of the cursor 2 along each of the three axes: X, Y, and Z. As described above, the 3 D movement of the cursor 2 could be signalled using other methods. A two dimensional cursor control device used in conjunction with another signal means could employed to produce 3D signals. Similarly, the alphanumeric keyboard 106 alone could be used as a 3D cursor control device. The processor logic, described below, applies the three axis movement to the vertices, faces, and edges comprising the cursor 2, computes a new projection to the screen

plane 5 for each cursor vertex, generates the shadow 3 that is associated with the cursor 2, and computes a new projection to the screen plane 5 for each shadow vertex.

The shadow 3 itself is a two dimensional shape, which can be displayed either as a group of polygons (shadow regions) or a group of edges (shadow edges). A shadow polygon 3 is cast by the opaque face of an object. An edge shadow (33 and 34) is cast by a translucent-faced object 32 or a non-solid object 31. The shadow 3 is associated with the cursor object 2 as determined by the position of the three dimensional cursor 2 object vertices stored in the object data structure. The vertices of the cursor 2 are translated to the two dimensional shadow plane 4 to produce shadow vertices. The shadow plane 4, as shown in the preferred embodiment, is perpendicular to the Y axis and located toward the bottom of the display screen 1 as shown in Figure 1. The shadow plane 4 is defined to simulate an infinitely large planar collimated light source. The light source is perpendicular to the Y axis; its light rays travel downward parallel to the Y axis. The shadow vertices define the association between the position of the shadow 3 and the position of the cursor 2. The computation of the shadow vertices, in the preferred embodiment, is described by the following formula for cursor vertex A:

$$a(x) = A(x)$$

$$a(y) = \text{Shadow Plane}(y)$$

$$a(z) = A(z)$$

Since the shadow vertices (e.g. a(x,y,z)) are derived from the cursor object vertices (e.g. A(x,y,z)), the shadow vertices will move as the cursor moves. The position of each of the shadow vertices is computed and stored in the object data structure with the movement of the cursor. For solid objects, pointers to the the shadow vertices

associated with each face are also retained in the object data structure.

Similar to objects, shadows are viewed by the user as an image projected to the screen plane 5 as shown in Figures 4 and 5. This computation is identical to the computation performed for the projection of the object vertices to the screen plane 5 as described above. The projection is used to determine the point in the screen plane 5 intersecting a line connecting each shadow vertex with the eye position 6. For example, as shown in Figure 5, a point q is the intersection of line ai and the screen plane 5. Line ah connects the object vertex with the coordinate space origin (h). The vector N 8 is a normal vector perpendicular to the screen plane 5. The location of the screen plane intersection point q is generated first by calculating the dot product of the vector ah with the screen normal vector N. Next, the dot product is computed for the vector ai with the screen normal vector N. The quantity of the dot product for ah divided by the dot product for ai is multiplied by the X,Y components of the ai vector and added to the coordinate components of the shadow vertex. The result is the point q in the screen plane 5 intersecting the line from the shadow vertex (a) to the eye position 6. The projection for each shadow vertex is computed as described in the following formulas for shadow vertex (a):

q(x) = (DotProduct(N, ah) / DotProduct(N, ai)) * ai(x) + a(x)

q(y) = (DotProduct(N, ah) / DotProduct(N, ai)) * ai(y) + a(y)

q(z) = Screen Plane(z)

The position of each of the other shadow vertices as projected to the screen plane 5 is computed in similar fashion and stored in the object data structure.

Movement of the cursor will cause corresponding movement of the screen plane projection points of the shadow vertices. For non-solid

objects, generation of the shadow is simply a matter of projecting the shadow vertices to the screen plane and drawing vectors between the screen plane projection points.

For solid objects, including those with either opaque or translucent faces, the generation of a shadow is not as straight forward. Not all faces of the solid object produce a shadow. Every edge of each face does not necessarily produce a distinct shadow. The present invention provides processing logic for determining which faces of the solid object cast shadows and which faces are non-shadowing. Since non-shadowing faces do not cast shadows, no shadow image needs to be drawn on the display screen for these faces. For shadowing faces, the shadow is drawn.

The normal vector associated with each face is used to determine if the face is shadowing or not. The dot product of the face normal vector with a vector pointing directly toward the light source 7 is computed. If the dot product is positive, the face will cast a shadow. If the dot product is zero, the face is parallel to the light source 7; therefore, no shadow is cast. If the dot product is negative, the face is hidden from the light source by another face (this is true only for closed solids); again, no shadow is cast.

The face normal vector is computed by taking the cross product of two edges of the face as in the following example for the face CAD in Figure 5:

Given vectors:

v1 = vector (AC), and

v2 = vector (AD), then:

Normal vector for face CAD = CrossProduct (v1, v2)

For each face of the solid, the normal vector is computed and tested against the dot product computed with the light vector as described above. If the face is shadowing (because the dot product is

positive), the shadow image is drawn on the screen plane as described below.

Since faces can be either opaque or translucent, two processing paths are provided to optimize the drawing of shadow images. For opaque shadowing faces 2, an opaque shadow region 3 will be produced. This shadow region is displayed by drawing an opaque polygon on the screen plane using the projected shadow vertices associated with this face. The position of the projected shadow vertices were computed and stored in the object data structure as described above. These vertices provide the corners of the shadow region in the screen plane. Many methods currently exist in the art of computer graphics for displaying filled regions on a display screen in this fashion.

For translucent shadowing faces 32, a translucent shadow region 34 is cast. This region is delineated on the display screen by drawing the edges of the face. Using the projected shadow vertices associated with this face, the edges are drawn on the screen plane as vectors using the projected shadow vertices as end points. Since each edge may be a component of more than one face, the potential exists that each edge may be drawn more than once. The present invention tests for this possibility and prevents the same edge from being drawn more than once. As each edge is drawn, a boolean flag (located in the object data structure for each edge) is set to a true value causing the subsequent drawing of the same edge to be inhibited.

The cursor 2 of the present invention may be activated using a function key or signal generation means 108 associated with the three dimensional cursor. The shadow 3 is displayed automatically with the display of the 3D cursor 2. Many other means of function activation may also be employed such as the entry of a special code

or command sequence from the alphanumeric keyboard 106 or an icon selection. An icon is small graphical symbol displayed to the user and identifiable as performing some function when selected. The present invention may also be activated through a software interface with the computer's operating system software or other application software. Using this method, the activation of the 3D cursor and the associated shadow can be independent of direct user action.

Upon activation, the 3D cursor 2 is presented to the user on the display screen 1 as shown in Figure 2. The cursor 2 is displayed along with the display of its associated shadow 3 on the shadow plane 4. The shape and orientation of the shadow 3 corresponds to the shape and orientation of the 3D cursor 2 in the manner described earlier. As the cursor control device 107 is moved, the 3D cursor 2 and shadow 3 move correspondingly. The 3D cursor 2 is allowed to move beyond the visible portion of the display screen 1. If the cursor 2 is moved above the top border of the display screen 1, the cursor 2 may disappear, but the shadow 3 is still displayed. If the cursor 2 is hidden behind another object on the display screen 1, the shadow 3 may still be visible. If the cursor 2 is moved below the shadow plane 4, the cursor 2 may be displayed but the shadow 3 is no longer displayed.

The shadow 3 itself is displayed to facilitate the movement of the cursor 2 in three dimensional space. The user is better able to judge height and distance when the cursor 2 is viewed in relation to the shadow 3. In the preferred embodiment, the shadow 3 appears to drop from immediately below the cursor 2 (this results from the direct derivation of the X and Z coordinate of the shadow vertices from the X and Z coordinates of the cursor vertices). In alternative embodiments, the shadow vertices may be defined indirectly from

the cursor vertices so that a shadow is cast at an angle 111 (as shown in Figure 11) from the cursor instead of directly below it. The angle may be applied to either or both of the X and Y axes. For example, assuming the light source 7 as shown in Figure 11 shining with parallel rays 110 (light vectors) , a vertex of the shadow may be computed by extending one of the light source rays 110 through the cursor vertex until the ray intersects the shadow plane 4.

Processing Logic for the Present Invention

The present invention includes computer program logic for the operation of the 3D cursor and corresponding shadow. This logic, executed by the processor 101, is described in the following section and in Figures 6-9. In addition to the computer resources described earlier, the present invention relies upon the availability of an operating system and system functions typically capable of displaying vectors on the display device 105. System functions for interfacing with the cursor control device 107 are also desirable, but not necessarily required. These resources are standard processing components known in the computer art.

When the system components of the present invention are first powered up, the operating system logic (typically residing in read only memory 103 and executed by the processor 101) takes control and initializes the other system components such as read/write memory 102, the display device 105, and the cursor control device 107. At the end of its initialization cycle or in response to a user command, the operating system may initiate the 3D cursor function. If initiated by the user via a function key activation, the 3D cursor program logic begins execution when the appropriate function key is activated. As described earlier, other means for activating the 3D cursor are available.

Once the 3D cursor program logic is activated, the processing flow begins as shown in Figure 6 at the box labeled "Start 3D Cursor and Shadow system" 601. First, the display system hardware is initialized 602. The next actions performed by the program logic is the display of all objects pre-defined in the object data structure. This data structure holds the data definition (vertices, faces, and edges) of all objects currently displayable in three dimensional space. One of these objects is the cursor.

The program logic begins a loop starting at box 603 in Figure 6. This loop initializes new image memory, updates the position of the objects in the object data structure, and displays the shadow for each object as the cursor control device is moved. First, the new image memory is erased 603. New image memory is a temporary buffer where the new display screen image is constructed. Each object for display on the display screen is added to new image memory as object processing is completed for that object. When the new image is completely built, new image memory is copied to display screen memory causing the display screen to show the new image. After the new image memory is erased 603, the shadow plane is drawn into new image memory 610. The shadow plane 4 is depicted in Figure 2. Next, a pointer is initialized to point to the top of the object data structure 611. Each object is retrieved from the object data structure 604, until there are no objects left to process 605. When all objects have been added to new image memory 607, program control is transferred to D in Figure 6. At D, the new image memory is copied to display screen memory 613. The image in display screen memory is then displayed on the display screen 612 and the loop for displaying objects begins again at box 603. When another object is found in the object data structure 606, the object

is retrieved 604 and processor control is transferred to A in Figure 7.

Starting at A in Figure 7, a boolean flag called the Show Shadow flag is initially set to false 719. The Show Shadow flag indicates whether the shadow for the current object should be displayed or not. Next, a new transformation matrix is generated 701. The generation of a transformation matrix is a well known prior art process. The transformation matrix defines the movement of the cursor control device along each of the three axes: X, Y, and Z. This movement is applied to each of the vertices in the object data structure in the next processing block 702. As the new 3D position of each vertex is computed, the Y component of the position is checked against the Y position of the shadow plane. If any vertex of the object is located above the shadow plane 703, the shadow for the object is signalled for display by setting the boolean Show Shadow flag to true 704; otherwise, the shadow will not be displayed (Show Shadow flag left false 711). For each vertex, the 3D position of the vertex is translated to the shadow plane 705 and projected to the screen plane for display 706. The translation to the shadow plane 705 is accomplished by setting the value of the vertex position Y coordinate to the value of the Y coordinate of the shadow plane as shown by the dashed lines Aa, Bb, Cc, and Dd (35) in Figure 5. If required, an angle is applied 111 to account for a displaced light source as shown in Figure 11. Projection to the screen plane 706 is performed by computing the position of the intersection point in the screen plane 5 with a vector connecting the object vertex and the eye position 6 as described earlier. The projection of each vertex to the screen plane is computed for both the object itself 706 (hence each object vertex) and the shadow of the object 707 (hence each shadow vertex). In Figure 5, the screen projection points of the

cursor vertices are shown as m, n, o, and p; the screen projection points for the cursor shadow vertices are q, r, s, and t. After projecting the object and shadow vertices to the screen plane, program execution loops back to process the next object vertex until all of the vertices for the object have been processed 708. If the eye position 6 is located below the shadow plane 709, the Show Shadow flag is set to false 717, since the shadow is not visible from an eye position below the shadow plane. If the eye position is above the shadow plane 716, the Show Shadow flag is left as previously set. Processing continues for the object faces and edges at B in Figure 8.

The faces for each object are processed as shown in Figure 8. If the object is non-solid 805, no faces exist; thus processing drops to C in Fig. 9. Since each face of the object does not necessarily cast a shadow, the processing loop from B to C is performed for each of the object faces defined in the object data structure. This loop is used to determine which of the faces are visible from the eye position and which faces cast shadow regions on the shadow plane. First, each face of the current object is retrieved from the object data structure 801. Next, the normal vector is computed for the face 802. The normal vector is computed by taking the cross product of two of the edges of the face as described above. The cross product yields a normal vector perpendicular to the face. The face normal vector is first used to determine if the face is visible from the eye position 810. This face normal vector is used to compute the dot product of the face normal vector with the negative of the vector pointing directly away from the eye position 6. If this dot product is greater than zero 811, then the face is visible. If this is the case, the face is drawn into the new image 813. If the face is not visible 812, the face is not drawn into the new image.

Next, the face shadow is processed. If the Show Shadow flag is true 815, then shadow processing for the face continues at E in Figure 8. If the Show Shadow flag is false 816, control skips to processing for the next face at B in Figure 8. For the shadow processing at E in Figure 8, the face normal vector is used to compute the dot product 803 of the face normal vector with the negative of the vector pointing directly away from the light source 110. If this dot product is greater than zero 807, then the face is shadowing. In this case, the shadow region for this face is drawn into the new display image 804. The shadow region is drawn using the screen-projected shadow vertices for this face computed earlier. The color or intensity of the shadow region is pre-defined in the object data structure. Shadow regions for faces with zero or negative dot products as computed above 806 are not drawn. When each of the faces are processed 809, edge processing begins at C in Figure 9.

The edges of each object require separate processing as shown in Figure 9. As described above, there are at least two types of objects: solid and non-solid. There are at least two types of solid objects: those with opaque faces and those with translucent faces. For solid objects with opaque faces, edge shadows do not need to be drawn, since the shadows are delineated by the colored or shaded shadow regions. In this case 906, program execution returns to D in Fig 6. For non-solid or translucent-faced solid objects, edge shadows need to be drawn as the object shadow would not otherwise be visible 907. Thus, a loop is initiated for each edge of the object defined in the object data structure 902 as shown in Figure 9. As each edge is retrieved from the object data structure, the edge is drawn into the new display image 912. If the Show Shadow flag is true 909, the shadow of the edge is also drawn into the display image 904. The

edge shadow is drawn using the projected shadow vertices computed earlier. When each edge has been processed 910, program execution returns to D in Fig. 6 where the object processing cycle begins again with a new cursor position.

Although this invention has been shown in relation to a particular embodiment, it should not be considered so limited. Rather it is limited only by the appended claims.

CLAIMS

1. In an interactive computer-controlled display system having a processor, a data display screen coupled to said processor, and a cursor control device coupled to said processor for interactively positioning a cursor on said display screen, a process for controlling the display of a shadow associated with said cursor, said process comprising the steps of:

generating and displaying a cursor on said data display screen, the position of said cursor being responsive to the three dimensional movement of said cursor control device, said cursor having at least one edge;

generating and displaying a shadow on said data display screen, said shadow being associated with said cursor, said shadow having an edge corresponding to an edge of said cursor projected on to a two dimensional shadow plane along a light vector, the position of said shadow being responsive to the three dimensional movement of said cursor control device.

2. The process of claim 1 wherein said cursor has a plurality of three dimensional vertices associated with said cursor and wherein said step of generating and displaying said cursor comprising:

maintaining cursor data associated with said cursor, said cursor data including a three dimensional position for each of said

plurality of three dimensional vertices associated with said cursor;

transforming said three dimensional position of said plurality of three dimensional vertices associated with said cursor to a new position based upon the three dimensional movement of said cursor control device;

drawing said cursor associated with said plurality of three dimensional vertices on said data display screen.

3. The process of claim 1 wherein said cursor has a plurality of three dimensional vertices and wherein said step of generating and displaying said shadow comprising:

generating shadow vertices by translating a three dimensional position for each of said plurality of three dimensional vertices associated with said cursor to a two dimensional shadow plane along said light vector, said shadow plane being substantially perpendicular to the vertical Y axis of the three dimensional space;

drawing said shadow associated with said shadow vertices on said data display screen.

4. The process of claim 2 wherein said computer-controlled display system further comprising a data storage means coupled to said processor, and wherein said process further comprising a step of maintaining cursor data, said step of maintaining cursor data comprising:

recalling, processing, and storing parameters associated with said plurality of three dimensional vertices associated with said cursor including the three dimensional position and configuration of said vertices;

recalling, processing, and storing parameters associated with the faces associated with said plurality of three dimensional vertices associated with said cursor;

recalling, processing, and storing parameters associated with edges associated with said faces;

5. The process of claim 2 wherein the shape and orientation of said cursor is responsive to the three dimensional movement of said cursor control device and wherein the shape and orientation of said shadow is responsive to the three dimensional movement of said cursor control device and wherein said step of drawing said cursor comprising;

projecting said plurality of three dimensional vertices associated with said cursor to a two dimensional screen plane, said screen plane being substantially perpendicular to the normal vector associated with said data display screen;

drawing the faces associated with said projected vertices of said cursor on said data display screen;

drawing the edges associated with said projected vertices of said cursor on said data display screen.

6. The process of claim 5 wherein said step of drawing said faces associated with said cursor comprising:

drawing the opaque faces associated with said plurality of three dimensional vertices associated with said cursor on said data display screen.

7. The process of claim 5 wherein said step of drawing said faces associated with said cursor comprising:

drawing the edges associated with said faces on said data display screen.

8. The process of claim 3 wherein said step of drawing said shadow comprising:

projecting said shadow vertices to a two dimensional screen plane, said screen plane being substantially perpendicular to the normal vector associated with said data display screen;

computing a normal vector for each of the faces associated with said cursor;

drawing shadow regions associated with said projected shadow vertices on said data display screen if no component of said normal vector for each of said faces associated with said cursor is parallel with said light vector and pointing toward said shadow plane;

drawing the shadow edges associated with said projected shadow vertices.

9. The process of claim 8 wherein said step of drawing said shadow regions associated with said shadow comprising:

drawing the opaque shadow regions associated with said projected shadow vertices on said data display screen.

10. The process of claim 8 wherein said step of drawing said shadow regions associated with said shadow comprising;

drawing the shadow edges associated with said shadow regions on said data display screen, each of said shadow edges drawn, if at all, only once.

11. The process of claim 1 wherein said shadow is not displayed if said cursor is moved below said shadow plane.

12. The process of claim 1 wherein said shadow is still displayed even if said cursor is not visible on said display screen.

13. An interactive computer-controlled display system comprising: a processor; a data display screen coupled to said processor; a cursor control device coupled to said processor for interactively positioning a cursor having at least one edge on said display screen;

a means for generating and displaying said cursor on said data display screen, the position of said cursor being responsive to the three dimensional movement of said cursor control device, said means for generating and displaying said cursor being coupled to said processor;

a means for generating and displaying a shadow on said data display screen, said shadow being associated with said cursor, said shadow having an edge corresponding to an edge of said cursor projected on to a two dimensional shadow plane along a light vector, the position of said shadow being responsive to the three dimensional movement of said cursor control device, said means for generating and displaying said shadow being coupled to said processor.

14. The device of claim 13 wherein said cursor has a plurality of three dimensional vertices and wherein said means for generating and displaying said cursor comprising:

means for maintaining cursor data associated with said cursor, said cursor data including a three dimensional position for each of said plurality of three dimensional vertices associated with said cursor;

means for transforming said three dimensional position of said plurality of three dimensional vertices associated with said cursor to a new position based upon the three dimensional movement of said cursor control device;

means for drawing said cursor associated with said plurality of three dimensional vertices on said data display screen.

15. The device of claim 13 wherein said cursor has a plurality of three dimensional vertices and wherein said means for generating and displaying said shadow comprising:

means for generating shadow vertices by translating a three dimensional position for each of a plurality of three dimensional vertices associated with said cursor to a two dimensional shadow plane along said light vector, said shadow plane being substantially perpendicular to the vertical Y axis of the three dimensional space;

means for displaying shadow regions associated with said shadow vertices on said data display screen.

16. The device of claim 14 wherein said computer-controlled display system further comprising a data storage means coupled to said processor, said means for maintaining cursor data associated with said cursor comprising:

means for recalling, processing, and storing parameters associated with said plurality of three dimensional vertices associated with said cursor including the three dimensional position and configuration of said vertices;

means for recalling, processing, and storing parameters associated with the faces associated with said plurality of three dimensional vertices associated with said cursor;

means for recalling, processing, and storing parameters associated with edges associated with said faces;

17. The device of claim 14 wherein said means for drawing said cursor comprising:

means for projecting said plurality of three dimensional vertices associated with said cursor to a two dimensional screen plane, said screen plane being substantially perpendicular to the normal vector associated with said data display screen;

means for drawing the faces associated with said projected vertices of said cursor on said data display screen;

means for drawing the edges associated with said projected vertices of said cursor on said data display screen.

18. The device of claim 17 wherein said means for drawing said faces associated with said cursor comprising:

means for drawing the opaque faces associated with said projected vertices of said cursor on said data display screen.

19. The device of claim 17 wherein said means for drawing said faces associated with said cursor comprising:

means for drawing said edges associated with said faces on said data display screen;

20. The device of claim 15 wherein said means for displaying said shadow regions comprising;

means for projecting said shadow vertices to a two dimensional screen plane, said screen plane being substantially perpendicular to the normal vector associated with said data display screen;

means for computing a normal vector for each of the faces associated with said cursor;

means for drawing shadow regions associated with said projected shadow vertices on said data display screen if no component of said normal vector for each of said faces associated with said cursor is parallel with said light vector and pointing toward said shadow plane;

means for drawing the shadow edges associated with said projected shadow vertices.

21. The device of claim 20 wherein said means for drawing said shadow regions associated with said shadow comprising:

means for drawing the opaque shadow regions associated with said projected shadow vertices on said data display screen.

22. The device of claim 20 wherein said means for drawing said shadow regions associated with said shadow comprising:

means for drawing the shadow edges associated with said shadow regions on said data display screen, each of said shadow edges drawn, if at all, only once.

23. The device of claim 13 wherein said shadow is not displayed if the cursor is moved below said shadow plane.

24. The device of claim 13 wherein said shadow is still displayed even if the cursor is not visible on said display screen.

25. In an interactive computer-controlled display system having a processor and a data display screen coupled to said processor, a process for controlling the display of a shadow associated with an object, said process comprising the steps of:

generating and displaying an object on said data display screen, said object having at least one edge;

generating and displaying a shadow on said data display screen, said shadow being associated with said object, said shadow having an edge corresponding to an edge of said object projected on to a two dimensional shadow plane along a light vector.

26. The process of claim 25 wherein said object has a plurality of three dimensional vertices associated with said object and wherein said step of generating and displaying said object comprising:

maintaining object data associated with said object, said object data including a three dimensional position for each of said plurality of three dimensional vertices associated with said object;

transforming said three dimensional position of said plurality of three dimensional vertices associated with said object to a new position;

drawing said object associated with said plurality of three dimensional vertices on said data display screen.

27. The process of claim 25 wherein said object has a plurality of three dimensional vertices and wherein said step of generating and displaying said shadow comprising:

generating shadow vertices by translating a three dimensional position for each of said plurality of three dimensional vertices associated with said object to a two dimensional shadow plane along a light vector, said shadow plane being substantially perpendicular to the vertical Y axis of the three dimensional space;

drawing said shadow associated with said shadow vertices on said data display screen.

28. The process of claim 26 wherein said computer-controlled display system further comprising a data storage means coupled to said processor, and wherein said process further comprising a step of maintaining object data, said step of maintaining object data comprising:

recalling, processing, and storing parameters associated with said plurality of three dimensional vertices associated with said object including the three dimensional position and configuration of said vertices;

recalling, processing, and storing parameters associated with the faces associated with said plurality of three dimensional vertices associated with said object;

recalling, processing, and storing parameters associated with edges associated with said faces;

29. The process of claim 26 wherein said step of drawing said object comprising;

projecting said plurality of three dimensional vertices associated with said object to a two dimensional screen plane, said screen plane being substantially perpendicular to the normal vector associated with said data display screen;

drawing the faces associated with said projected vertices of said object on said data display screen;

drawing the edges associated with said projected vertices of said object on said data display screen.

30. The process of claim 29 wherein said step of drawing said faces associated with said object comprising:

drawing the opaque faces associated with said plurality of three dimensional vertices associated with said object on said data display screen.

31. The process of claim 29 wherein said step of drawing said faces associated with said object comprising:

drawing the edges associated with said faces on said data display screen.

32. The process of claim 27 wherein said step of drawing said shadow comprising:

projecting said shadow vertices to a two dimensional screen plane, said screen plane being substantially perpendicular to the normal vector associated with said data display screen;

computing a normal vector for each of the faces associated with said object;

drawing shadow regions associated with said projected shadow vertices on said data display screen if no component of said normal vector for each of said faces associated with said cursor is parallel with said light vector and pointing toward said shadow plane;

drawing the shadow edges associated with said projected shadow vertices.

33. The process of claim 32 wherein said step of drawing said shadow regions associated with said shadow comprising:

drawing the opaque shadow regions associated with said projected shadow vertices on said data display screen.

34. The process of claim 32 wherein said step of drawing said shadow regions associated with said shadow comprising;

drawing the shadow edges associated with said shadow regions on said data display screen, each of said shadow edges drawn, if at all, only once.

35. The process of claim 25 wherein said shadow is not displayed if said object moves below said shadow plane.

36. The process of claim 25 wherein said shadow is still displayed even if said object is not visible on said display screen.

37. In an interactive computer-controlled display system having a processor substantially as hereinbefore described with reference to the accompanying drawings.

Published 1991 at The Patent Office, State House, 66/71 High Holborn, London WC1R 4TP. Further copies may be obtained from Sales Branch, Unit 6, Nine Mile Point, Cwmfelinfach, Cross Keys, Newport, NP1 7HZ. Printed by Multiplex techniques ltd, St Mary Cray, Kent.